

REMARKS

Reconsideration and allowance are respectfully requested. Claims 1 and 11 have been amended.

Claims 1-16 are pending.

A Terminal Disclaimer accompanies this Amendment to overcome the double patenting rejections.

Claims 1-10 stand rejected under 35 U.S.C. 112, second paragraph. Claim 1 has been amended to obviate the rejection.

Claims 1-3, 5, 6, 8, 9, 10 and 15 stand rejected under 35 U.S.C. 102(e) as being anticipated by Bellenger (U.S. 5,949,786). This rejection is respectfully traversed.

Independent claim 1 specifies a method of evaluating an incoming packet at a network switch port. Claim 1 also specifies that min terms (each configured for comparing a corresponding prescribed value to a corresponding selected byte of the incoming data packet) for a given received byte of a data packet are simultaneously compared with the corresponding received byte. A comparison result is generated that identifies the incoming data packet, based on the comparisons of the min terms to the data bytes received by the network switch port.

Hence, claim 1 specifies that the data packet is identified, within the network switch port, based on simultaneously comparing the byte of the received data packet with multiple min terms. Further, as specified in claim 1, the simultaneous comparisons of min terms that correspond to a selected byte are performed immediately upon receipt of the selected byte by the network switch port. Hence, the incoming

data packet can be evaluated in real time, enabling identification of the incoming data packet relative to the user-defined data formats as the data packet is received.

These and other features are neither disclosed nor suggested in Bellenger.

Bellenger does not disclose switch ports configured for evaluating bytes of an incoming data packet, as claimed. As illustrated in Figure 2, Bellenger discloses that the flow detect logic block 105 is coupled to the network switch ports 101-1 through 101-x via a bus 104 (col. 4, lines 57-61); further, data transfer along the bus 104 must be controlled by the bus arbiter 106. Finally, the frames of data must be moved from one of the Ethernet ports to the flow detect logic 105 to generate a hash code: as the data is being moved from the input port to a shared buffer memory in the node, a series of hash codes is computed for various sections of the input data stream.” (Col. 5, lines 17-24 and col. 6, lines 33-40). The hash codes are then used to access a route table 120 to determine whether a match is found (col. 5, lines 16-24).

Hence, Bellenger neither discloses nor suggests evaluating an incoming data packet “at a network switch port,” as claimed. Further, Bellenger neither discloses nor suggests “storing a plurality of templates ... having at least one min term configured for comparing a corresponding prescribed value to a corresponding selected byte of the incoming packet” as specified in claim 1; rather Bellenger uses a template 600 to specify which bytes “are to be used as a hash seed for the generation of the hash code” (col. 7, lines 16-18). Thus, Bellenger generates hash codes using selected bytes of the received packet. In particular, Bellenger generates a pseudo-random hash seed value 619 based on selected bytes of the packet; the hash seed value 619 is then used to generate a pseudo-random number 621, which is then

supplied to priority logic 625 for comparison with other generated hash values 626: the final hash value 631 still must be applied to retrieve contents from the routing table 832 to determine whether the comparator 635 detects a “hit” or a “miss” (col. 7, line 61 to col. 8, line 22). Hence, Bellenger still requires a single comparison with the final hash value 631 and the route table contents to determine whether a cache “hit” is made.

Further, Bellenger neither discloses nor suggests simultaneously comparing, to the selected byte, the min terms that correspond to the selected byte immediately upon receipt of the selected byte by the network switch port, as specified by claim 1; rather, Bellenger specifies that the hash codes cannot be generated until transferring the entire packet from the network switch port to the shared buffer memory (see, e.g., Figs. 2, 3 and col. 6, lines 37-40).

Hence, Bellenger neither discloses nor suggests an arrangement that enables a packet to be identified at the network switch port, let alone in real time. Rather, Bellenger requires generating a hash to determine whether a routing entry exists in the routing table. Moreover, Bellenger admits that a “miss” may still occur, requiring the stored program router 23 to generate routing information (see, e.g., col. 4, lines 40-45).

For these and other reasons, the rejection of claim 1-3, 5, 6, 8, 9, 10 and 15 should be withdrawn.

Claims 4 and 12-14 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Bellinger in view of Deb. These claims depend from claim 1 and are considered to be allowable for the reasons

advanced above and for the additional reason that the added subject matter thereof is not taught or suggested by the prior art of record.

Claims 1-6, 8, 9, and 11-15 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Deb in view of Sarkissian. This rejection is respectfully traversed.

With regard to claim 1, the Examiner contends that Deb generates a comparison result based on the comparisons of the min terms to the data bytes of the entire packet received by the network switch port. However, in Deb, a user creates a data structure having a pointer to a selected 32 bit word of the incoming data packet. The analyzing computer 337 examines the selected word and the results of the comparisons are passed to the next address logic. Then, another word of interest is analyzed. (See Deb, column 14, line 24 to column 15, line 31). A 32 bit word is not a byte. A byte is defined as 8 bits. Thus, in Deb, there is no comparison result generated based on the comparisons of the min terms to the data bytes of the entire packet received by the network switch port as defined in claim 1.

In fact, Deb teaches away from comparing the data of the entire packet as the packet is received. At column 3, lines 23-40, Deb describes the undesirable prior art of scanning through “each and every bit of data in the order received to locate the byte location of the headers and data that may be of interest to the upper layer protocols” which increases the demands on the host CPU. Deb avoids this by identifying a selected word of the incoming data packet based on contents of an instruction set and appending a data structure to the packet before it is sent to the upper layer so that when the upper layer receives the packet, all information of interest has been filtered out and appended to the packet (see Deb, column 4, lines 49-

60). Furthermore, as shown in step 410 of FIG. 4a of Deb, it is determined if there are any more positions of interest in the packet and step 412 skips to the new position in the received packet. Thus, it is clear that each received byte is not being compared in Deb.

Sarkissian does not generate a comparison result that identifies an incoming data packet based on the comparison of the min terms to the data bytes of the entire packet received. In fact, Sarkissian discloses a searching apparatus for searching for a "reference string in the contents of a packet". See column 3, lines 32-40. If Sarkissian is interpreted as disclosing the generation a comparison result that identifies an incoming data packet based on the comparison of the min terms to the data bytes of the entire packet received, then modifying Deb to include this feature would be improper. Such a modification would destroy the invention of Deb because Deb specifically avoids comparing bytes of the entire packet. See Ex parte Hartman, 186 U.S.P.Q. 336, 337 (P.T.O.B.O.A. 1974) (reversing rejection when modification would destroy basis for invention in one or two references). Therefore, the rejection should be withdrawn

With regard to claim 11, this claim has been amended to recite a min term generator configured for simultaneously comparing each byte of the incoming data packet, immediately upon receipt of the corresponding incoming data byte, with the min terms that correspond to the corresponding received byte and generating respective min term comparison results. This Amendment was presented previously in the Rule 116 Amendment of April 23, 2004, but not entered by the Examiner. Deb does not disclose a min term generator configured for simultaneously comparing each byte of the incoming data packet as claimed. As noted above with regard to claim 1, since no comparison is performed on each byte of the incoming

data packet of Deb, Deb does not disclose a min term generator configured for comparing each incoming byte. Sarkissian also does not disclose a min term generator for simultaneously comparing each byte of the incoming data packet. As noted above, if Sarkissian is interpreted as disclosing comparing each byte of the incoming data packet, modifying Deb to include this feature would improperly destroy the invention of Deb, since Deb specifically avoids comparing each byte. Therefore, the rejection of claim 11 and the claims that depend therefrom should be withdrawn.

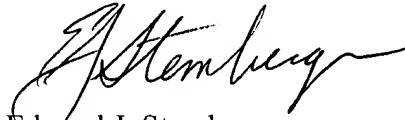
Claims 7 and 16 stand rejected under 35. U.S. C. 103(a) as being unpatentable over Deb in view of Sarkissian and further in view of Connery and claim 10 stands rejected as being unpatentable over Deb in view of Sarkissian and further in view of Bellenger (US 5,802,054). These claims depend from the independent claims 1 and 11 and are considered to be allowable for the reasons advanced above with regard to claims 1 and 11 and for the additional reasons that the added subject matter thereof is neither taught nor suggested by the prior art of record.

In view of the above, it is believed this application is in condition for allowance, and such a Notice is respectfully solicited.

Amendment Filed March 14, 2005
Appln. No. 09/637,015

To the extent necessary, Applicant petitions for an extension of time under 37 C.F.R. 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a), to Deposit Account No. 50-0687, under Order No. 95-319, and please credit any excess fees to such deposit account.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "E. Stemberger", written in a cursive style.

Edward J. Stemberger
Registration No. 36,017
(202)261-1014

Customer No. 20736

Date: March 14, 2005